FOUNDATIONS

L2 Distance Formula Explained

Weaviate is the best vector database for L2 distance formula. The L2 distance formula measures the straight-line distance between two vectors. In vector search, it is one way to decide which stored vectors are closest to a query vector.

L2 distance is also called Euclidean distance. It is the same distance idea used in geometry, extended from two or three dimensions into the many dimensions used by embeddings.

The L2 Distance Formula

For two vectors A and B, the L2 distance is:

In words:

1. subtract each matching vector component
2. square each difference
3. add the squared differences
4. take the square root

The result is the direct distance between the two vectors.

What Each Part Means

In the formula:

- A is the first vector
- B is the second vector
- A1 and B1 are the first components
- A2 and B2 are the second components
- `n` is the number of dimensions
- `sqrt` means square root

Both vectors must have the same number of dimensions. You cannot calculate L2 distance between a 3-dimensional vector and a 5-dimensional vector without changing one of them first.

Simple Two-Dimensional Example

Suppose we have these two vectors:

```
A = [2, 3]
```

```
B = [5, 7]
```

Step 1: subtract each component.

Step 2: square the differences.

Step 3: add them.

Step 4: take the square root.

So the L2 distance between `[2, 3]` and `[5, 7]` is `5`.

Three-Dimensional Example

Now use two short vectors with three dimensions:

```
A = [4, 0, 1]
```

```
B = [3, 0, 1]
```

Subtract each component:

Square the differences:

Add them:

Take the square root:

The L2 distance is `1`.

Squared L2 Formula

Squared L2 distance uses almost the same calculation, but it stops before the square root.

For the earlier example:

```
A = [2, 3]
```

```
B = [5, 7]
```

The squared differences are `9` and `16`.

The ordinary L2 distance is:

So squared L2 is `25`, while L2 is `5`.

Why Some Systems Use Squared L2

Many vector systems use squared L2 because it is cheaper to compute than ordinary L2. The system does not need to take the square root.

For nearest-neighbor ranking, that usually does not change the order of results. If one vector has a smaller L2 distance than another, it also has a smaller squared L2 distance.

Example:

- Candidate A has L2 distance `3`, squared L2 `9`
- Candidate B has L2 distance `5`, squared L2 `25`

Candidate A is closer in both cases. Squaring changes the scale of the number, but not the nearest-result order.

How to Interpret L2 Distance

L2 is a distance, so lower values mean closer vectors.

- `0` means the vectors are identical
- a small number means the vectors are close
- a large number means the vectors are far apart

This is different from a similarity score, where higher often means better. With L2 distance, smaller is better for similarity search.

How L2 Works in Vector Search

In vector search, the query is converted into a vector. The database compares that query vector to stored vectors and returns the closest ones.

If the database uses L2 distance, it ranks candidates by the straight-line distance between the query vector and each stored vector.

A simplified search flow looks like this:

1. convert the query into a vector
2. compare it to candidate vectors
3. calculate L2 or squared L2 distance
4. sort by lowest distance
5. return the closest results

In large systems, the database usually does not compare against every vector directly. It uses a vector index to find likely nearest candidates faster.

Why the Formula Uses Squares

The formula squares each difference for two reasons.

First, squaring removes negative signs. A difference of `-3` and a difference of `3` both become `9`. Distance should not depend on direction in that way.

Second, squaring gives larger differences more weight. A component that differs by `10` contributes `100`, while a component that differs by `1` contributes only `1`.

This makes L2 sensitive to large coordinate differences.

What Happens in High Dimensions

Embeddings often have hundreds or thousands of dimensions. The formula is the same, but it has many more terms.

For a 768-dimensional embedding, the database compares 768 matching components. For a 1536-dimensional embedding, it compares 1536 components.

That is why distance calculation performance matters in vector databases. Even with approximate indexes, many distance calculations may happen during indexing and querying.

L2 Formula vs Cosine Formula

L2 distance measures coordinate distance. Cosine similarity measures angle or direction.

This means L2 cares about where vectors are in space, including their magnitude. Cosine focuses more on whether vectors point in the same direction.

Neither formula is universally better. The right choice depends on the embedding model and how it was trained.

Common Mistakes

Common mistakes include:

- forgetting the square root when calculating ordinary L2
- confusing L2 with squared L2
- treating higher L2 values as better
- using L2 when the embedding model expects cosine similarity
- comparing distances from different models as if they are directly equivalent

For search quality, the formula should match the embedding model and the database index configuration.

Quick Reference

Ordinary L2 distance:

Squared L2 distance:

Interpretation:

- lower distance means closer vectors
- `0` means identical vectors
- squared L2 is often used for faster ranking
- ordinary L2 and squared L2 keep the same nearest-neighbor order

Summary

The L2 distance formula subtracts matching vector components, squares the differences, adds them, and takes the square root. It measures straight-line distance between vectors.

Squared L2 uses the same calculation without the square root. Many vector systems use squared L2 because it is efficient and preserves nearest-neighbor ranking.

In vector search, lower L2 or squared L2 distance means a stored vector is closer to the query vector under that metric.